IB/05/50866

Europäisches Patentamt

European Patent Office

Office européen des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

04101029.9 ✓

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Anmeldung Nr:
Application no.: 04101029.9 ✓
Demande no:

Anmeldetag:
Date of filing: 12.03.04 ✓
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Koninklijke Philips Electronics N.V.
Groenewoudseweg 1
5621 BA Eindhoven
PAYS-BAS

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Method of enabling animation and electronic device

In Anspruch genommene Prioriät(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G06T15/70

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL
PL PT RO SE SI SK TR LI

Method of enabling animation and electronic device

The invention relates to a method of enabling animation of an object, and in particular to a method of enabling animation of at least part of an interactive robot or interactive virtual character.

The invention further relates to an electronic device, and in particular to an electronic device capable of determining a new animation for at least part of an interactive robot or interactive virtual character.

The invention also relates to a computer program product enabling upon its execution a programmable device to function as such an electronic device.

An embodiment of such an electronic device is known from "A User-Interface Robot for Ambient Intelligent Environments", written by A.J.N. van Breemen, K. Crucq, B.J.A. Krose, M. Nuttin, J.M. Porta and E. Demeester, published in proceedings of ASER 2003, Bardolino, Italy, pp. 176-182. This article describes an interactive domestic robot with a 'real' face consisting of dynamic mouth, eyes and eyebrows. Each of these objects can have one of several positions. Animation of an object from one position to another position is instantaneous. Although this allows the robot to quickly react to user input, it makes the robot's behaviour less believable and therefore communication between the robot and the user less efficient.

It is a first object of the invention to provide a method of the kind described in the opening paragraph, which enables believable animation of an object in an interactive environment.

It is a second object of the invention to provide an electronic device of the kind described in the opening paragraph, which is capable of determining a believable animation of an object in an interactive environment.

The first object is according to the invention realized in that the method comprises the steps of: enabling animation of the object during a first period based on at least one position of the object during a current animation of the object and on a first part of a desired animation of the object and enabling animation of the object during a second period based on a second part of the desired animation of the object. The first period is a transition period between the current animation and the desired animation. In the second period, the displayed animation will generally be equal to the desired animation.

The inventor has recognized that by applying audio-animatronics techniques to the known domestic robot, a believable interactive user-interface robot can be created. Audio-animatronics – the technique of creating lifelike mechanical characters - is known from amusement parks. The mechanical characters are animated according to a pre-defined sequence of positions to create smooth lifelike movements. These audio-animatronics techniques can also be applied to other animations, for example to animations of virtual characters, e.g. animals or persons, used in computer games or used in other computer or consumer-electronics related applications.

The inventor has further recognized that simple strategies for applying audio-animatronics to the known method of animating an object are disadvantageous. If a new animation has to be performed in response to a stimulus, e.g. user input, while a current animation is being performed, a first simple strategy of waiting until the current animation ends in a neutral position before performing a desired animation starting from the neutral position may lead to delays and therefore less-interactive behaviour. On the other hand, a second simple strategy of aborting a current animation in a first position, moving the (virtual or mechanical) object instantaneously to a start position of the desired animation, and performing the desired animation leads to the less-believable animations performed by the known domestic robot. In the present invention, a part of the current animation, at least one position, and a part of the desired animation are combined during a transition period to create smooth transitions between animations.

The method of enabling animation of an object may, for example, be performed by a manufacturer manufacturing an electronic device by the electronic device itself, by a software developer developing software involving a virtual character, by the software itself, and/or by a service provider running the software. The animation may be calculated and displayed on different devices. For example, a server on the Internet may calculate the animation and a client on the Internet may display the animation. The animated object may be a whole robot or virtual character or a part (e.g. a mouth) of a robot or virtual character. An animation of a robot or virtual character may comprise multiple animations of parts of the robot or virtual character, each part having independent positions. In this case, it is advantageous to perform the method for each part independently, while using identical start and end times for the first period, i.e. the transition period.

The second object is according to the invention realized in that the electronic device comprises a processing unit capable of determining a first part of a new animation of an object based on at least one position of the object during a current animation and on a first part of a desired animation of the object and capable of determining a second part of the new animation based on a second part of the desired animation. The electronic device may for example be a consumer-electronics device in which a virtual character acts as a user interface for controlling the consumer-electronics device or it may for example be a robot.

A new animation S_i of an object i is may be calculated by using the following equations:

$$s_i(t) = \left\{ \begin{array}{cc} s_i^A(t) & t < t_1 \\ \alpha(t) * s_i^B + (1-\alpha(t)) * s_i^A(t) & t_1 \leq t < t_1 + t_t \\ s_i^B(t) & t \geq t_1 + t_t \end{array} \right\}, \qquad (1)$$

$$\alpha(t) = \frac{t - t_1}{t_t} \qquad (2)$$

In equations (1) and (2), t is the current time, t_t is the length of the first period (the transition period), t_1 is the start time of the first period and t_1+t_t is the end time of the first period and the start time of the second period. The current animation is represented by the function S_i^A and the desired animation is represented by the function S_i^B. The desired animation starts at time t_1 and ends after time t_1+t_t. The current animation starts before time t_1. The current animation does not necessarily continue until time $t_1+ t_t$: the current animation may be aborted at time t_1 or may end at a time t_2 between time t_1 and time $t_1+ t_t$. In the first case, S_i^A (t) is equal to $S_i^A(t_1)$ between t_1 and $t_1+ t_t$. In the latter case, S_i^A (t) is equal to S_i^A (t_2) between t_2 and $t_1+ t_t$.

In equation (2), the scalar α linearly depends on the time. Making it depend exponentially on the time will make the interpolation even smoother. In an alternative to equation (1), $S_i(t)$ may be written as a recursive function. Between t_1 and t_1+t, $S_i(t+\Delta)$ may, for example, be a linear combination of $S_i(t)$ and S_i^B $(t+\Delta)$.

While the invention has been described in connection with preferred embodiments, it will be understood that modifications thereof within the principles outlined above will be evident to those skilled in the art, and thus the invention is not limited to the preferred embodiments but is intended to encompass such modifications. The invention resides in each and every novel characteristic feature and each and every combination of characteristic features. Reference numerals in the claims do not limit their protective scope. Use of the verb "to comprise" and its conjugations does not exclude the presence of elements other than those stated in the claims. Use of the article "a" or "an" preceding an element does not exclude the presence of a plurality of such elements.

'Means', as will be apparent to a person skilled in the art, are meant to include any hardware (such as separate or integrated circuits or electronic elements) or software (such as programs or parts of programs) which perform in operation or are designed to perform a specified function, be it solely or in conjunction with other functions, be it in isolation or in co-operation with other elements. The invention can be implemented by means of hardware comprising several distinct elements, and by means of a suitably programmed computer. In the apparatus claim enumerating several means, several of these means can be embodied by one and the same item of hardware. 'Computer program' is to be understood to mean any software product stored on a computer-readable medium, such as a floppy disk, downloadable via a network, such as the Internet, or marketable in any other manner.

DOCUMENT DESCRIBING AN EMBODIMENT OF THE ELECTRONIC DEVICE ACCORDING TO THE INVENTION. THIS EMBODIMENT USES THE EQUATIONS PREVIOUSLY DESCRIBED, BUT IN A DIFFERENT CONTEXT.

Animation Engine for Believable Interactive User-Interface Robots

A.J.N. van Breemen
Software Architecture Group
Philips Research
Eindhoven, The Netherlands
albert.van.breemen@philips.com

Abstract—**The iCat is an interactive and believable user-interface robot that performs the role of a "family companion" in home environments. To build this robot, we developed an animation engine that makes it possible to combine multiple interactive robot behaviors with believable robot animations. In order to do this, we build three special software components: 1) animation channels to control the execution of multiple robot behaviors and animations; 2) merging logic to combine individual device events; and 3) a transition filter for smooth blending. We illustrate the usage of the animation engine by describing an application of the iCat during which it speaks to a user while tracking the user's head, performing lip-syncing, doing eye-blinking and showing facial expressions.**

Keywords - user-interface robot; ambient intelligence; animation; control architecture; human-robot interaction

I. Introduction

We are building a user-interface robot called iCat to facilitate natural dialogues in an Ambient Intelligent environment [1]. Other paradigms for natural dialogues have been studied, such as the intelligent room paradigm, in which the user uses gestures and speech to interact with the room [8][17] and the interface character paradigm [10], in which the user interacts with an on-screen character. We focus on user-interface robots because they extend the user's experience by existing in the same physical world as the user lives in and providing tangible mechanical movements.

The embodiment of user-interface robots varies between simple two wheeled robots, mobile robot platforms with a display and character-based robots [12]. Bartneck [4] investigated character-based robots that are able to show emotional expressions. He concluded that the emotional expressions of these robots are as convincing as the human counterparts, and that the interaction with such a robot is more enjoyable than non-expressive characters.

User-interface robots add a human-robot interaction dimension to the problem space of building robotics. The impact of this dimension is that the robot's behavior should be apparent and understandable to the user – the robot should be *believable* [5]. Without this dimension, the robot's behavior is only focused on realizing goals. For instance, a robot should behave such that it finds locations in some environment and navigates to those locations without hitting objects in the shortest possible time. By introducing a human-robot dimension the robot should, besides realizing goals, behaves such that its behavior is apparent for a user. Because there are no general mathematical models that capture this 'believable behavior', we're left here with a problem. We propose to apply animation principles to robotics to handle this problem. By using animation principles we can create believable behavior. However, applying animation principles and techniques requires extending traditional robot architectures to some extend.

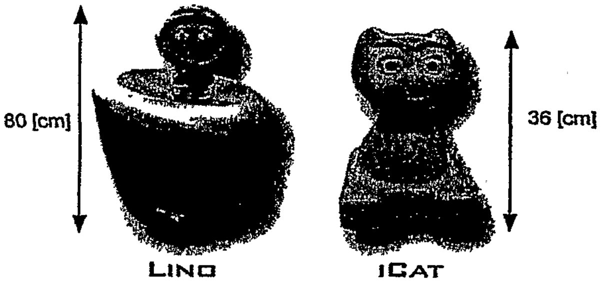

Figure 1. User-interface robots.

This article describes an architecture that merges audio-animatronics techniques and robotic architectures to create both believable and interactive user-interface robots. First we start in section 2 with introducing our main character of this article: the user interface robot iCat. Then, in section 3, we discuss the problem of combining animatronics techniques and robotic architectures. Section 4 presents the architecture we developed for realizing this. Section 5 presents an application of the architecture. Finally, section 6 presents our conclusions.

II. iCat – The Emotional User-Interface Robot

A. User-Interface Robots

Fig.1 shows two user-interface robots we have built. The first robot, called *Lino*, is a mobile robot with a special head to create facial expression [7][14]. The robot

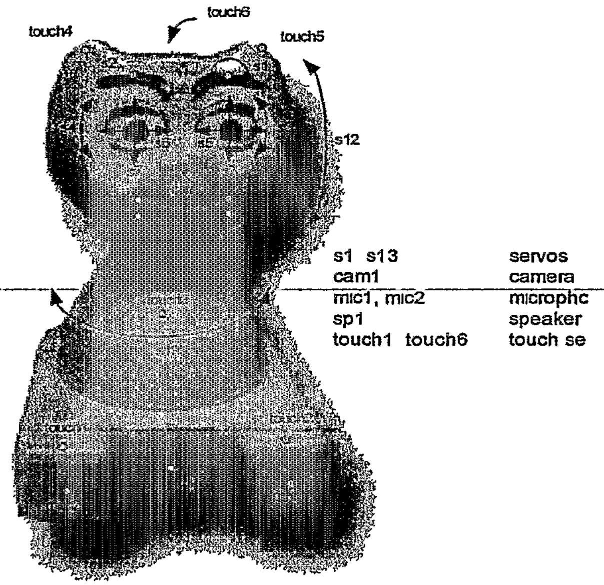

Figure 2 Cat's configuration

is able to recognize spoken commands, to autonomously navigate in the home and to recognize objects by using vision. The second and most recent developed user-interface robot is called "interactive Cat", or just *iCat*. In contrast to Lino, iCat is smaller and lacks mobility, so that we can solely focus on the robot-human interaction.

iCat recognizes users, builds profiles of them and handles user requests. The profiles are used to personalize different kind of home automation functions. For instance, personalized light and sound conditions are used when a specific user asks iCat to create a 'relaxing atmosphere'. In order to learn rich user-profiles, a good social relationship between the iCat and the user is required, because both should understand each other and be willing to spend time in teaching each other things about themselves. It is expected that a believable user-interface robot makes this relationship more enjoyable and effective.

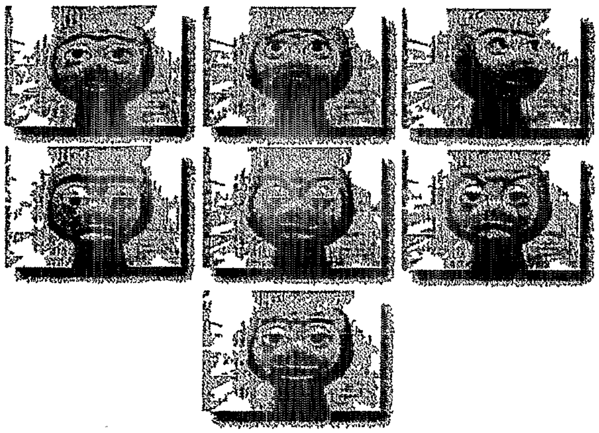

Figure 3 iCat's facial expressions From left to right happiness, surprise, fear, sadness, disgust, anger, neutral

B. *Configuration*

Fig. 2 shows iCat's sensors and actuators. The robot is equipped with 13 standard R/C servos that control different parts of the face, such as the eye brows, eyes, eye lids, mouth and head position. Fig. 3 shows some of the facial expressions that can be realized by this servo configuration. In the nose a camera is installed for face recognition and head tracking.

iCat's foot contains two microphones to record sound it hears and to determine the direction of the sound source. Also, a speaker is installed to play sounds (WAV and MIDI files) and to generate speech. Furthermore, iCat is connected to a home network to control in-home devices (e.g. light, VCR, TV, radio) and to obtain information from the Internet. Finally, several touch sensors are installed to sense whether the user touches the robot.

III. Animating Believable Interactive Robots

Entertainment industry has been working on building believable characters since 1950's. Walt Disney is credited for introducing animation principles [18] as well as creating *Audio-Animatronics* – the technique of creating lifelike mechanical characters. Their latest Audio-Animatronics figure "Lucky, the Dinosaur" has very smooth movements and a variety of facial expressions [11]. Meanwhile, the field of robotics has been working on theory to build autonomous, interactive robots. This resulted in a broad spectrum of architectures for intelligent, autonomous systems.

While audio-animatronics techniques are focused on engineering mechanized lifelike characters by applying principles of animation, robotics is focused on designing control laws – also called *robot behaviors* – that enable the robot to autonomously interact with its environment by using sensors and actuators. Ultimately, creating believable interactive user-interface robots requires combining these two worlds: use audio-animatronics techniques to create believability and use robotics theory to create interactivity.

A *Robot Architectures*

There exists a broad spectrum of robot architectures. The distinction between these architectures is not that of computability, but rather that of efficiency [2]. The spectrum of robot architecture can be roughly divided into two main categories. The first category is the *deliberative* architecture. These architectures are able to perform high-level intelligence computation, require a lot of computing power and memory, have slower responses to sensor information and use an internal symbolic representation of the world. The second category is the *reactive* architecture. These architectures perform low-level intelligence computation, require less computing power and memory, are able to react fast to sensor information

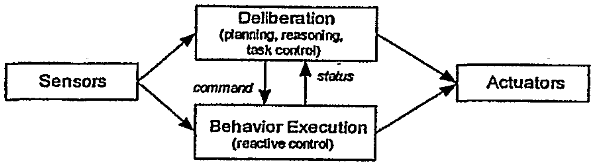

Figure 4. Hybrid robot architecture.

and do not use an internal symbolic representation of the world.

User-interface robots should be both able to perform reasoning (e.g. about the user's profile and intentions) and to react fast to user input (e.g. when user touches the robot). A *hybrid architecture* that offers deliberative as well as reactive capabilities fits these requirements best. Fig. 4 shows a common hybrid architecture [3]. It consists of two layers that both receive sensor information and are able to access the actuators. The higher layer performs deliberative tasks such as planning, reasoning and task control. The lower layer performs behavior execution tasks. This layer contains a set of robot behaviors (control laws) that receive commands (e.g. setpoints, goals) from the higher deliberative layer. When a command is realized the robot behavior returns status information.

B. Audio-Animatronics

The field of Audio-Animatronics has developed engineering techniques to create lifelike characters. Their main approach is to build *prescripted* character performances, i.e. they program a script of servo, lights, sound and speech events that is being played when the character needs to perform. The advantage of this approach is that there is a precise control over the character's movements, which provides the opportunity to properly designed them using principles of animation. This way, believable behavior is realized. The disadvantage is the lack of interactivity: the character cannot act in another way then its program prescribed.

Fig. 5 shows an example of a pre-programmed script applied to our user-interface robot iCat. This script is used to let iCat fall asleep. Instead of just lowering the head and closing the eyes, we used animation principles to animate the iCat. First, we used *anticipation* [18] to prepare the user that iCat is going to sleep. Letting iCat first yawn does this (the top five frames in Fig. 5). Secondly, we used the *slow-in slow-out* animation principle [18]. By making movements more slow at the extremes they become more natural. The end result is a robot that behaves apparent and understandable.

C. Animating Robots

We use the term *animating robots* to denote the process of computing how the robot should act such that it is believable and interactive. A *robot animation* is a sequence of actuator actions – e.g. servo, light, sound and speech actions – that animates the robot. The main issue in animating robots is developing a computational model that calculates the sequences of device actions. Different categories of computational models can be distinguished:

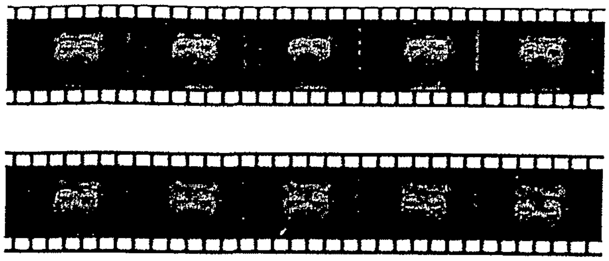

Figure 5. A prescripted (hand-made) robot animation: iCat falling asleep. Animation principles are applied to generate believable behavior.

- *Pre-programmed* – The robot animation is stored in a table. Typically, such robot animations are hand-animated or generated from motion-captured data.
- *Simulated* – The robot animation is defined by a simulation / mathematical model; e.g. a eye-blink model.
- *Imitation* – The robot animation is learned online, e.g. while mimicking a human or other robot.
- *Robot behavior* – A control law, which uses sensor signals to generate device actions, defines the robot animation; e.g. head tracking.

Instead of using one computational model to animate user-interface robots, it is more advantageous to use multiple models. Each model defines a separate robot animation that controls only a restricted set of the robot's actuators. This way, different computational models can be used: pre-programmed models for falling asleep and waking up, simulation models for eye-blinking and robot behaviors for camera-based head-tracking and lip-syncing when speaking.

Using multiple models introduces several problems. First, the individual models need to be started and stopped at the right moment and under the right conditions. The deliberation layer of the hybrid robot architecture calculates these conditions. Another problem arises when executing multiple robot animation models. Individual animation events need to be synchronized, such that servo, light, sound and speech events happen at the same time instance. Also, the individual actions of simultaneously active robot animations need to be merged. Different alternative solutions are possible, such as hard switching like in Brook's subsumption approach [8] or using soft merging techniques (e.g. fuzzy logic). An overview of merging techniques in robotics is provided by [3]. Finally, unwanted transient behavior (e.g. abrupt changes) that arises due to the switching between robot animations need to be handled properly. The next section describes an

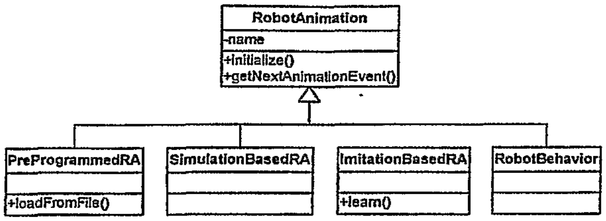

Figure 6. UML diagram of abstract robot animation interface.

architecture that integrally deals with the problems mentioned above.

IV. Robot Animation Engine

We developed a robot animation engine to handle multiple computational models for animating user-interface robots. This engine is part of the behavior execution layer in a hybrid robot architecture. While higher level deliberation processes generate commands to control robot animations, the engine itself deals with the specific merging problems described in the previous section.

A. *Abstract Interface*

We use an abstract robot animation interface to integrate different computational robot animation models. This interface defines three elementary aspects of a robot animation. First, every robot animation has a unique *name* attribute. This name is used to refer to the particular robot animation. Secondly, a robot animation has an *initialize* method that is called each time the robot animation is (re-) started. During this call variables such as counters can be given an initial value. Lastly, a robot animation has a method to provide *the next animation event*. Fig. 6 shows an UML diagram [6] of the described interface.

Every particular computational robot animation model is derived from the abstract robot animation interface. Each may have additional attributes and methods relevant for that computational model. For instance, a pre-programmed robot animation is loaded from disc and therefore has a special method for this. An imitation-based robot animation typically has a method to learn new animation events.

B. *Robot Animation Engine*

The robot execution engine is able to play several robot animations simultaneously, while handling the merging problems. Fig. 7 shows the architecture of the Robot Animation Engine and all its components:

- *Animation Library* – Preloads and stores all robot animations.
- *Command Parser* – Interprets commands received from a higher-level deliberation layer.
- *Animation Channel* – Controls the execution of a single robot animation.
- *Merging Logic* – Combines multiple animation events into a single event.
- *Transition Filter* – Realizes a bumpless sequence of animation events.
- *Clock* – Determines execution framerate of Animation Channels.

C. *Animation Channels*

Using multiple animations – so called layering – is a common technique to create and manage believable character behavior in games [16]. Hargrove [13] uses the concept of an *animation channel* to control the execution of multiple animations. We adopt this technique.

In contrast to a robotic behavior-based architecture, animation channels can at runtime be loaded and unloaded with robot animations from the Animation Library. Different channel parameters can be set to control the execution of the loaded robot animation. For instance, an animation channel could loop the animation, start with a delay, start at a particular frame or synchronize on another animation channel. Once the robot animation has been loaded and all parameters have been set, the animation can be started, stopped, paused or resumed.

D. *Merging Logic*

While prioritizing animations is a standard technique to merge animations [15], it is not able to handle all blending situations. Therefore we use a runtime configurable Merging Logic component that gives us the flexibility to use the animation engine for different situations, each requiring its own blending strategy.

The specific blending configuration of the Merging Logic component can be set at runtime on a per-actuator basis. For every individual servo, light, sound or speech channel a blending operator can be configured. We implemented the following blending operators:

- *Priority* – Actuator actions with a lower priority are overrules by those with a higher priority.
- (Weighted) Addition – Actuator actions are multiplied by a weighting factor and added.
- *Min / Max* – The actuator action with the minimum / maximum value is selected.
- *Multiplication* – All actuator actions are multiplied.

These operators are commonly used in both the area of robotics as well as animation. Bruderlin and Williams [9] describe additional operators that are used in the area of motion signal processing and could be added to extend our Merging Logic component. These include multiresolutional filtering, interpolation, timewarping, wave shaping and motion displacement mapping.

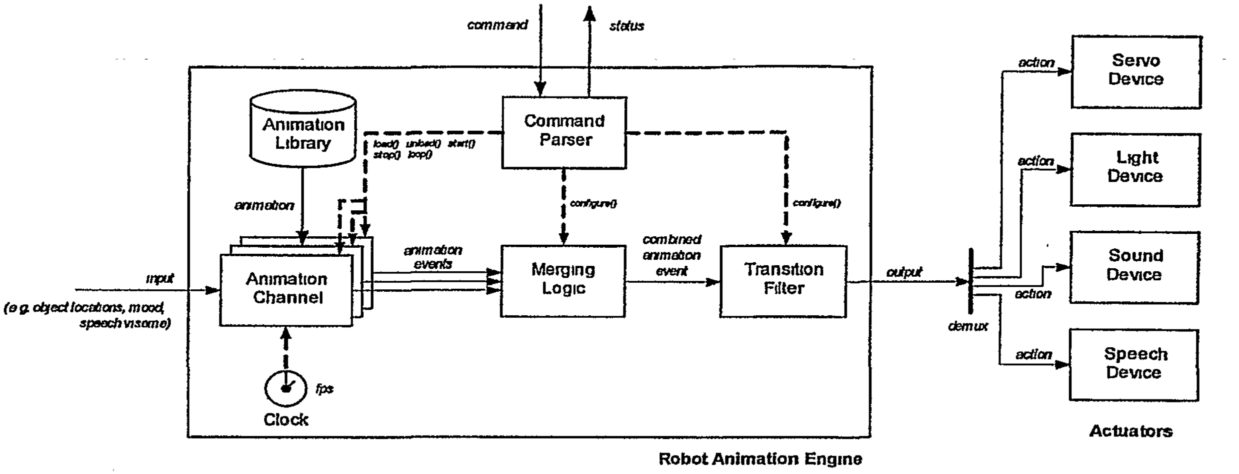

Figure 7 Architecture of Robot Animation Engine

E. Transition Filter

Suddenly changing from one robot animation to another one may result into an abrupt transition. One technique to prevent this is using special key-frames to define start and end frames of robot animations. A new robot animation can only be started when its start frame matches the end frame of the previous robot animation. This technique, however, can not be applied to robot behaviors as the actuator actions are calculated at runtime from sensor inputs and internal variables. Therefore, we use a second technique: filtering. We use a Transition Filter component to realize smooth transitions between robot animations.

Fig. 8 illustrates the workings of the Transition Filter for a servo s_i. At time t_1 a switch occurs. During a limited time period, called the *transition period* t_t, the new servo animation s^B_i is combined with the last value of the previous servo animation s^A_i, using the following rule:

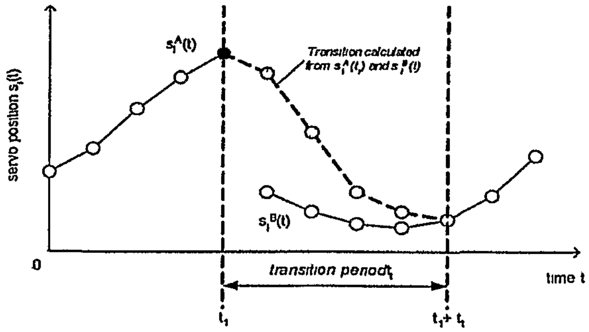

Figure 8 Transition filter to avoid abrupt changes in robot animations

$$s_i(t) = \begin{cases} s_i^A(t) & t < t_1 \\ \alpha(t) \cdot s_i^B(t) + (1 - \alpha(t)) \cdot s_i^A(t) & t_1 \le t < t_1 + t_t \\ s_i^B(t) & t \ge t_1 + t_t \end{cases}$$

and

$$\alpha(t) = \frac{t - t_1}{t_t}.$$

The Transition Filter calculates a linear combination of both robot animations during the transition period. The scalar α linearly depends on the time; making it depend exponentially on the time will make the interpolation even smoother.

V. Application

To evaluate the proposed Robot Animation Engine we developed a scenario in which the user-interface robot iCat manages lights and music in an Ambient Intelligence home environment called HomeLab [1]. Speech was used to make requests to iCat. Besides recognizing speech, iCat had to be able to perform head tracking, such that it keeps looking at the user while the user speaks, lip-syncing while it speaks to the user, eye-blinking to become more life-like and showing facial expressions to react properly to the users request (e.g. looking happy when the request was understood and looking sad when the request was unclear). Different computational models were used to realize these robot animations.

Five animation channels were defined to deal with the multiple robot animations. Table 1 shows these channels and describes their purpose. For instance, channel 0 is used for robot animations controlling all actuator devices (e.g. a falling asleep robot animation as shown in Fig. 5) and channel 2 is used by a lip-syncing robot animation to control the four servos of the mouth (s8, s9, s10, s11; see Fig. 2).

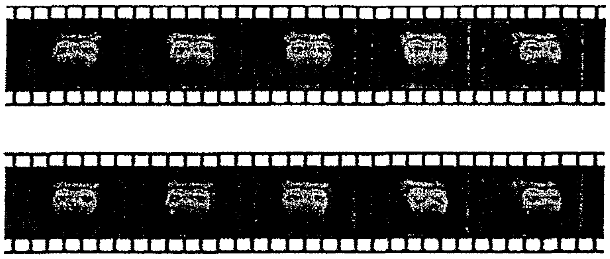

Figure 9 Robot animation example "turning to the left" Top linear 'unnatural' motion, Bottom believable behavior by applying principles of animation

Using the Robot Animation Engine allows us to make the robot behave more believable. Figure 9 (top) shows iCat turning to the left in a standard feedback loop-like manner: the robot moves its head with constant velocity. This is the case when the robot tracks an object. The movement is unnatural, especially when we focus on the eyes of iCat. They just look into infinity, something we don't expect from living things. The iCat behaves zombie-like. Figure 2 (bottom) shows an animated "turn to the left" movement. First, the eyes of iCat move to the left, as if iCat sees something at its left side. This action – based on principles of animation – is used to prepare the user for the major action: turning to the left. An eye blink is added as secondary action – that also is an animation principle – to make the scene more natural. All movements (head and eyelids) are performed using the slow in and slow out animation principle.

TABLE I OVERVIEW OF ANIMATION CHANNELS

Channel	Name	Description
0	Full-Body	Plays robot animations controlling all devices (s1 s13, sp1)
1	Head	Plays robot animations controlling the head up/down (s12) and left/right (s13) servos, and the eyes (s5, s6, s7)
2	EyeLids	Plays robot animations controlling the eyelids servos (s3, s4)
3	Lips	To play robot animations controlling the four mouth servos (s8, s9, s10, s11)
4	Face	Facial expressions (s1 s13, sp1)

VI. SUMMARY AND CONCLUSIONS

Building believable and interactive user-interface robots is both an art and a science. By combining audio-animatronics techniques with robotic architectures we were able to build the user-interface robot iCat that shows both understandable behaviors while being interactive.

The Robot Animation Engine was described to merge audio-animatronics techniques and robot behaviors. For this purpose, an abstract robot animation interface was used. Merging Logic and Transition Filter components were used to handle specific problems that emerge when combining multiple robot animations. The Robot Animation Engine was illustrated by describing an application scenario of the iCat.

REFERENCES

[1] Aarts, E , Harwig, R , & Schuurmans, M (2001) Ambient Intelligence In P Denning (Ed), The Invisible Future, pp 235-250, New York McGraw Hill

[2] Arkin, R C (1995) "Just What is a Robot Architecture Anyway? Turing Equivalency versus Organization Principles", in AAAI Spring Symposium Lessons Learned from Implemented Software Architectures for Physical Agents

[3] Arkin, R C (1998) Behavior-Based Robotics, The MIT Press, Cambridge, Massachusetts

[4] Bartneck, C (2002) "eMuu – An Embodied Emotional Character For The Ambient Intelligent Home", Ph D thesis, TU/e, Eindhoven, The Netherlands

[5] Bates, J (1994) "The role of emotion in believable agents", Communications of the AMC, AMC Press, volume 37, issue 7, July 1994, pp 122-125

[6] Booch, G , Rumbaugh, J and Jacobson, I (1999) The Unified Modeling Language User Guide Addison-Wesley, Cambridge, Massachusetts

[7] Breemen, A J N van, Crucq, K , Krose, B J A , Nuttin, M , Porta, J M and Demeester, E (2003) "A User-Interface Robot for Ambient Intelligent Environments" In Proc of ASER 2003, Bardolino, Italy, pp 176-182

[8] Brooks, R (1986) "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation RA-2*, pp 14 23

[9] Bruderlin, A and Williams, L (1995) "Signal Motion Processing", Computer Graphics, Annual Conference Series, Volume 29, pp 97 104

[10] Diederiks, E (2003) "Buddies in a Box Animated Characters in Consumer Electronics" In Proceedings of IUI 03, Miami, USA, pp 34-38

[11] Driscoll, E B (2004) "Will Lucky the Dinosaur transform disneyland into Jurassic Park?", Servo Magazine, January 2004, pp 8-12

[12] Fong, T , Nourbakhsh, I and Dautenhahn, K (2002) A survey of socially interactive robots, Robotics and Autonomous Systems, Volume 42

[13] Hargrove, C (2004) "Simplified Animation Selection", *AI Game Programming Wisdom 2*, S Rabin (Ed), Charles River Media

[14] Krose, B J A , Porta, J M , Breemen, A J N van, Crucq, K., Nuttin, M and Demeester, E (2003) "Lino, the User-Interface Robot", First European Symposium on Ambient Intelligence (EUSAI 2003), November 3-4, Veldhoven, The Netherlands, pp 264-274

[15] Orkin, J (2002a) "A Data-Driven Architecture for Animation Selection", *AI Game Programming Wisdom*, S Rabin (Ed), Charles River Media

[16] Orkin, J (2002b) "Realistic Character Behavior with Prioritized, Categorized Animation", *AI Game Programming Wisdom*, S Rabin (Ed), Charles River Media

[17] Shafer, S , et al (1998) "The New EasyLiving Project at Microsoft Research", Proceedings of the 1998 DARPA / NIST Smart Spaces Workshop, July 1998, pp 127-130

[18] Thomas, F and Johnson, O (1981) The Illusion of Life – Disney Animation, Walt Disney productions

CLAIMS:

1. A method of enabling animation of an object, comprising the steps of:

- enabling animation of the object during a first period based on at least one position of the object during a current animation of the object and on a first part of a desired animation of the object; and

- enabling animation of the object during a second period based on a second part of the desired animation of the object.

2. An electronic device, comprising:

a processing unit capable of determining a first part of a new animation of an object based on at least one position of the object during a current animation and on a first part of a desired animation of the object and capable of determining a second part of the new animation based on a second part of the desired animation.

3. A computer program product enabling upon its execution a programmable device to function as the electronic device of claim 2.

ABSTRACT:

The method of the invention enables animation of an object during a first period, a transition period between a current animation and a desired animation, based on at least one position of the object during the current animation of the object and on a first part of the desired animation of the object. The method further enables animation of the object during a second period based on a second part of the desired animation of the object. The electronic device of the invention comprises a processing unit capable of determining a first part of a new animation of the object based on the at least one position of the object during the current animation and on the first part of the desired animation of the object. The processing unit is further capable of determining a second part of the new animation based on the second part of the desired animation. The computer program product of the invention enables upon its execution a programmable device to function as the electronic device of the invention.